SQUID Proxie-Server

- Kurzakte –

Server:

Installation	
- mittels Yast aus Paketgruppe Produktivity/Networking/Web/Proxie - Achtung: bei Deinstallation werden Cache und Logdateien nicht mit entfernt	
Besondere Kennzeichen	
• Proxyserver für die Protokolle HTTP, HTTPS und FTP	
Konfiguration Normalzugang	
/etc/squid/squid.conf	• alleinige Konfigurationsdatei • auskommentierte Zeilen enthalten die Default-Werte des Servers und sollten nicht entkommentiert werden (stört Serverbetrieb) • zum Ändern sollten jeweilige Zeilen kopiert und entkommentiert werden • mögliche Zugriffsbeschränkungen: Quelle, Ziel, Domain, Zeit, Browser
/var/cache/squid/	• Cache-Verzeichnis; muss existieren und für Benutzer squid beschreibbar sein
Start als Stand-Alone-Dienst	
/usr/sbin/squid squid –z squid –k reconfigure squid –N –d 1 –D squid –k parse	• manueller Start des Daemons durch Aufruf des Binaries • Anlegen der Verzeichnisstruktur für den Cache und Beenden (als user squid) • sendet ein HUP-Signal an Squid (-k "Schlüsselwort für Signal") • manueller Start des Daemons zu Testzwecken (-D ohne DNS-Check) • überprüft Konfigurationsdatei auf syntaktische Fehler
/etc/init.d/squid rcsquid (SuSE)	• start | stop | restart | reload | status - Skript startet/stoppt squid manuell • legt beim ersten Start den Cache-Bereich an (1.024 Verzeichnisse) • Achtung: ohne eingerichteten Nameserver läuft hier nichts!!!
cd /etc/init.d/rc5.d ln –s /etc/init.d/squid S99squid ln –s /etc/init.d/squid K01squid	• Einrichten des automatischen Starts des Skripts • Anlegen von symbolischen Links als Start- und Stop-Skript in die entsprechenden Runlevel-Verzeichnisse
Logging	
/var/log/squid/	• Log-Verzeichnis; muss existieren und für Benutzer squid beschreibbar sein
/var/log/squid/access.log	• protokolliert Clientzugriffe laut Direktive cache_access_log • Skript zum Umwandeln des Linux-Zeitstempels in eine lesbare Form: while read LINE; do echo -n $(date -d "1970/01/01 $(echo $LINE \| cut -f1 -d.) seconds"); echo "$LINE" | cut -c15- ; done • Parameter emulate_httpd_log on, macht Zeitangabe menschenlesbar
/var/log/squid/cache.log	• allgemeine Meldungen zum Serverbetrieb
/var/log/squid/store.log	• Meldungen zu allen lokalen Speichervorgängen des Proxies • kann mit "cache_store_log none " abgeschaltet werden, da es kaum brauchbare Analysetools für diese Datei gibt
calamaris sarg	• in Perl geschriebenes Auswertungstool `cat / var/squid/logs/access.log | calamaris –a [-F html]`
Erreichbarkeit eines DNS-Servers	
/etc/resolv.conf	• Eintrag eines Provider-DNS-Servers bei Einwahl mit dyn. Adressvergabe • Falls ein solcher nicht existiert, kann ein caching-only DNS auf der gleichen Maschine eingerichtet werden, erreichbar mit der Adresse 127.0.0.1 hier erfolgt die Weiterleitung der Anforderungen über root.hint o. forwarder
/etc/named.conf	• Eintrag des Name-Servers des Providers unter forwarders für schnellere Anfragen (nicht zwingend)
Firewalleinrichtung	
Regeln	• auf dem Proxy den Zugriff der Rechner aus dem lokalen Netz erlauben • „vor internem Netz schützen“ muss deaktiviert werden
Dokumentation	
man: squid (8), squid_ldap_auth (8), squid_ldap_group (8) und im Verzeichnis /usr/share/doc/packages/squid/	

Client:

- Eintrag des Proxy-Servers samt Port in den entsprechenden Web-Browser
- unter Linux: Erstellen und Exportieren der Variable http_proxie=http://proxy.domain.de:Port

alternativ:

Auf dem Internet-Router die transparente Nutzung des Proxies konfigurieren
iptables -t nat -A PREROUTING -p tcp -i ethx --dport 80 -j DNAT --to ip.des.gateways:3128

Inhalt der /etc/sqiud/squid.conf

administrative Parameter

cache_mgr E-Mail-Account **Default: webmaster**
erreichbare email-Adresse bei Cache-Problemen
cache_effective_user User **Default: squid**
cache_effective_group Gruppe **Default: -**
nach dem Start mit Root-Berechtigung ändert Squid seine Identität (euid und egid) laut dieser Direktive.
ftp_user E-Mail-Adresse **Default: Squid@**
Account für die Einwahl bei anonymen FTP-Servern
http_port [Hostname:|IP-Adresse:]Portnummer Default: 3128
Socketadresse, auf der Squid nach Anfragen von Clients lauscht
Ohne die Angaben von Hostnamen bzw. IP-Adressen gilt der angegebene Port für alle Interfaces. Ansonsten kann diese Anweisung für verschiedene Netzwerkkarten verschiedene Ports benutzen. Dazu muss für jeden gewünschten Port eine separate Befehlszeile angegeben werden.
visible_host_name Name **Default: -**
bei Mitteilungen nach aussen wird nicht der kanonische name (uname –n) gesendet, sondern der hier eingetragene
Dieser muss in einem Ressource Record des zuständigen Nameservers eingetragen sein (squid IN 1D CNAME gustav)

Steuern der Caching-Funktion

cache_mem GrößeMB|KB **Default: 8 MB**
Größe des reservierten RAM-Bedarfs für Cache- Objekte (Gesamtbedarf meist das 3fache, zB. Verwaltungsdaten, DNS,)
cache_dir Typ Pfad Limit FirstLevelDir SecondLevelDir Default: ufs /var/cache/squid 100 16 256
legt den Plattencache fest (ist zwingend zum Betrieb des Proxies erforderlich)

- Typ Angabe des Dateisystemtyps (ufs ist der gebräuchlichste)
- Pfad Verzeichnispfad zum Cache mit SQUID Schreibrecht (muss vorhanden sein!!!)
- Limit maximale Größe der Gesamtsumme der gecachten Objekte (max. 80% von df)
- FirstLevelDir Anzahl der in erster Ebene liegenden Verzeichnisse (meist 16)
- SecondLevelDir Anzahl der Verzeichnisse in jedem FirstLevelDir (üblich sind 256)

cache_swap_low Zahl **Default: 90**
cache_swap_high Zahl **Default: -**
sanftes (low) bzw. aggressives Putzen der Platte bei Erreichen dieser Grenzen, um Cache-Größe nicht zu überschreiten
maximum_object_size Zahl KB **Default: 4096 KB**
minimum_object_size Zahl KB **Default: 0 KB**
erlaubte Minimal- und Maximalgröße von zu cachenden Objekten (default sind 0 und 4096)
minimum_object_size_in_memory Zahl KB **Default: -**
Maximalgröße von zu cachenden Objekten im RAM, hier lohnt es sich zu experimentieren
quick_abort_min Zahl KB **Default: 16 KB**
quick_abort_max Zahl KB **Default: -**
quick_abort_pct Prozentzahl **Default: -**
Entscheidungskriterien zur Fortsetzung des Downloads zu Cachingzwecken, falls ein benutzer die laufende Webübertragung im Browser abbricht

DNS

dns_nameservers IP-Adresse
Eintrag des Nameservers des ISP.
nur dann notwendig, wenn die Resolver-Bibliothek des TCP/IP-Stacks nicht über diesen Nameserver informiert ist, was über die /etc/resolv.conf geschehen würde. Der Nameserver ist damit nicht systemweit bekannt, sondern nur dem Squid.

Zugriffssteuerung mit ACLs (Zugangsgewährung anhand der Netzwerkadressen und Benutzerinformationen)

1. Definition von ACLs:

allgemeine Form der Bedingungen und Bedingungstypen:

```
acl ACLName ACLTyp Element1 Element2 ...
acl ACLName ACLTyp "Dateiname"
```

- ACLName eindeutiger Listenname
- ACLTyp Listentyp
- Element1 Inhalt
- Dateiname Hier wird der Inhalt aus einer Datei ausgelesen; " " sind hier zwingend zur Abgrenzung von Elementen

beziehen sich mehrere ACL-Direktiven auf denselben Namen, werden sie additiv gewertet:

```
acl netz1 src 192.168.1.1/32 }
acl netz1 src 192.168.1.2/32 }   hat exakt dieselbe Bedeutung wie
acl netz1 src 192.168.1.1/32 192.168.1.2/32
```

die wichtigsten ACL-Listen-Typen

arp MAC-Adresse ...
Liste von MAC-Adressen als Argument
```
acl VERSUCH arp 00:00:11:22:33:44   00:33:44:55:66:77
```
src IP-Adresse/Netzmaske ...
Adressliste der Clients, die SQUID nutzen dürfen
Netzmasken entweder in Kurzform (z.B. /24) oder klassische (z.B. /255.255.255.0
```
acl VERSUCH src 192.168.1.0/24   192.168.100.7/255.255.255.255
```
dst IP-Adresse/Netzmaske ...
Adressliste der Server, auf die SQUID zugreifen soll (Zielserver). Diese müssen durch Rückwärtsauflösung eines DNS-Servers verifizierbar sein.

port Portnummer ...
Liste von Portnummern des Ziel-Servers, entweder als Portliste oder -bereich (mit Bindestrich getrennt)
```
acl VERSUCH port 80 81 443 1234-1299
```
srcdomain Domainname ...
Domainnamen der Clients, die SQUID benutzen wollen
Domainnamen sollten mit einem Punkt beginnen, etwa .mydomain.de
```
acl VERSUCH srcdomain .mydomain.de .myseconddomain.de
```
dstdomain Domainname ...
Liste von Domainnamen von (Web-)Servern an, auf die SQUID zugreifen soll

time [Wochentag] h1:m1-h2:m2 ...
Liste von Zeitfenstern
Der optional angegebene Wochentag wird mit den folgenden Buchstaben ausgedrückt: M=Montag, T=Dienstag, W=Mittwoch, H=Donnerstag, F=Freitag, A=Samstag, S=Sonntag.
```
acl VERSUCH time 09:30-17:00    # Jeden Tag von 9:30 bis 17:00 Uhr
acl VERSUCH time S 00:00-23:59  # Sonntags immer
```
url_regex [-i] Regulärer_Ausdruck ...
Liste von regulären Ausdrücken, die URLs beschreiben. –i schaltet case-Sensitivität ab
```
acl VERSUCH url_regex http://.* ftp://.*
```
urlpath_regex [-i] Regulärer_Ausdruck ...
Liste von regulären Ausdrücken, die nur den Pfadanteil einer URL beschreiben (ohne Protokollheader und Hostnamen)
```
acl VERSUCH urlpath_regex -i \.gif$ \.jpg$ \.png$
```
proto Protokollnamen ...
Liste der von durch SQUID unterstützte Protokollnamen (was in der URI vor dem :// steht)
```
acl VERSUCH proto FTP HTTP
```
browser [-i] Regulärer_Ausdruck ...
Liste von regulären Ausdrücken, die den Browser des Clients beschreiben (Wert von User-Agent im Header).
```
acl VERSUCH browser [mM]ozilla.*
```
proxy_auth Username ...
Liste von Usernamen, die über einen externen Prozess (PAM, LDAP) authentifiziert wurden.
Das Schlüsselwort REQUIRED bedeutet, dass jeder existierende Benutzer gemeint ist.
```
acl VERSUCH proxy_auth peter michael gabi
```
proxy_auth_regex [-i] Regulärer_Ausdruck ...
analog proxy_auth, nur mit regulären Ausdrücken statt Usernamen

2. Direktiven zur Vergabe von Zugriffsrechten auf die einzelnen Listen

allgemeine Form der Regeln und Regellisten:
Direktive {allow|deny} [!] ACLName [[!] ACLName]

- Direktive
- allow|deny Direktive wird erlaubt oder verboten
- ! die angegebene Regel wird nicht auf diese ACL angewendet
- ACLName mehrere ACLNamen werden durch logisches UND verknüpft

die wichtigsten Direktiven

http_access
Erlaubt oder verbietet Clients den Zugriff auf den HTTP-Port von SQUID. Die erste passende Regel wird verwendet. Ist diese Direktive nicht vorhanden wird der HTTP-Zugriff allen erlaubt. (Policy: erlaube allen alles)
Wurden http_access Regeln formuliert, treffen aber nicht zu, wird die letzte http_access Regel in ihrem Wahrheitswert umgedreht und als Standard-Einstellung verwendet. Hat also die letzte http_access-Anweisung ein deny, so ist die Voreinstellung für alle nicht zutreffenden Fälle ein allow und umgekehrt.

no_cache deny
Definiert, welche Anfragen direkt an den Server geschickt werden, ohne den Cache zu benutzen.

redirector_access
Die URL der Anfrage wird durch die Redirector-Schnittstelle von Squid geleitet.

auth_param
Legt fest, welche Programme zur Authentifizierung von Benutzern herangezogen werden (PAM, LDAP, …).

Beispiele

http_access allow localhost	erlaubt Zugriff aller Clients, die auf dem Squid-Host laufen
http_access allow meinnetz zeit	alle Benutzer aus "meinnetz" dürfen in der "zeit" aufs Internet zugreifen
http_access allow deinnetz !zeit	alle Benutzer aus "deinnetz" dürfen in der "zeit" nicht zugreifen
http_access deny all	Auffangregel am Ende

Logging

logfile_rotate Zahl **Default: 0**

beträgt der Wert 10 werden 9 Backup-Files erstellt
Defaultwert 0 schaltet das Rotating aus, falls logrotate genutzt wird.

Authentifizierung (Zugangsgewährung anhand der Benutzerinformationen)
proxy_auth

Eine beispielhafte Minimalkonfiguration:

```
cache_mem 8 MB
cache_dir ufs /var/squid/cache 100 16 256

acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl all src 0.0.0.0/0.0.0.0
acl meinnetz src 192.168.10.0/255.255.255.0 # ACL Rechner des LANs
acl keinmp3 urlpath_regex -i \.mp3     # erfasst alle mp3-Dateien
acl gesperrt  dstdomain  www.google.de # erfasst die Webseite von Google

# die Reihenfolge ist wichtig! Squid bewertet nur den ersten gültigen Treffer
http_access deny keinmp3            # verbietet Download von mp3-Dateien
(http_access allow checkpw all)     # Zugriff für angemeldete user, s. unten
http_access allow manager localhost # erlaubt manager nur von localhost
http_access deny manager all        # alle anderen Manager verbieten
http_access deny gesperrt
http_access allow meinnetz          # Zugriff für alle Rechner unseres LANs
http_access deny all                # alle anderen haben keinen Zugriff

icp_access allow allowed_hosts
icp_access deny all
```

deutsche Sprache einstellen
den Symlink errors in /etc/squid auf /usr/lib/squid/errors/German setzen

Bsp.: Authentifizierung

```
auth_param basic program /usr/sbin/ncsa_auth /etc/squid/passwd
#auth_param basic program /usr/sbin/pam_auth # alternativ für Systembenutzer
#auth_param basic program /usr/sbin/smb_auth -W ARBEITSGRUPPE  # Samba-Accounts
#auth_param basic program /usr/sbin/squid_ldap_auth -b         # LDAP-Accounts
#"ou=people,dc=your,dc=domain" ldapserver
auth_param basic program children 5     # aktive Instanzen
auth_param basic realm Internetzugang   # Authentisierungsbereich
auth_param basic credentialsttl 2 hours# wann erneute Nachfrage bei ncsa_auth
acl checkpw proxy_auth REQUIRED         # ACL checkpw gilt für alle User
http_access allow checkpw               # Direktive (Reihenfolge beachten!)
```

Erstellen der Paswortdatei /etc/squid/passwd

```
pro Benutzer eine Zeile:    username:passwordhash
Erstellen eines md5-Passworts z.B. mit grub-md5-crypt
Test mit:       echo paul password | ncsa_auth /etc/squid/passwd
```

Squid Defaultwerte anzeigen:

```
linux:~/scripte # squid -v
Squid Cache: Version 2.5.STABLE6
linux:~/scripte # grep -A 1 Default /etc/squid/squid.conf | egrep -v
"(Default|--|none)" | grep -v "#    "
# http_port 3128
# ssl_unclean_shutdown off
# icp_port 3130
```

Content-Filtering mit Squidguard

Aufrüsten von Squid-Guard als eigenständigem Zusatzprogramm für eine Application-Level-Firewall

redirect_program /usr/bin/squidGuard [-c /etc/squidguard/squidGuard.conf]
redirect_children 5 # Anzahl der von Squid zu startenden Redirector-Prozesse

Squid-Guard-Basiskonfiguration

```
# /etc/squid/squidGuard.conf
# hier liegen weitere Regeldateien, z. B. schwarze und weiße Domainlisten
 dbhome /etc/squid
# Verzeichnis des Logfiles, das z. B. Hinweise auf fehlerhafte Konfig-Zeilen enthält
 logdir /var/log/squid

# Regelsätzen der Verbindungsquellen anhand von Benutzernamen (Schlüsselwort user)
 src fullaccess {
   user mommy daddy
 }

# alle Kinder stehen in der datei /etc/squid/kiddies
src lessaccess {
   userlist kiddies
 }

# Regelsatz der Verbindungsquellen anhand von IP-Adressen
 src rechner {
   userlist myusers
 }

# Regelsätze mit Zielen, Umleitung einer heißen Domain auf eine Seite mit Verbotsnachricht
# Logging verbotener Zugriffe in /var/log/squid/sex.log
 dest porno {
   domain somethinghot.com
   redirect http://www.meine-domain.de/verboten.html
   log sex.log
 }

# weiße Domainlisten in /etc/sqiud/restrict/ *, Squid liest diese beim reload ein
dest restrict {
   domainlist restrict/domains
   urllist restrict/urllist
   expressionlist restrict/expressionlist
 }

# Erstellen eines finalen Regelsatzes
# alle Benutzer von fullaccess dürfen zu allen Zieladressen durch
# Die Benutzer von lessaccess dürfen nur auf die Domänen von restrict zugreifen
# Die Grundeinstellung default untersagt jeden externen Verkehr
acl {
   fullaccess {
     pass all
   }
   lessaccess {
     pass restrict none
     pass !porno all
   }
   default {
     pass none
     redirect http://my-domain.com/blocksite.php ?url=%u&user=%i&client=%a
   }
 }
```